Attorney Docket No. 1293.1916

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yang-lim CHOI

Application No.: Unassigned | Group Art Unit: Unassigned

Filed: September 16, 2003 | Examiner: Unassigned

For: METHOD OF MANAGING METADATA

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant submits herewith a certified copy of the following foreign application:

Korean Patent Application No. 2003-13002

Filed: March 3, 2003

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: September 16, 2003 By: _____

Michael D. Stein
Registration No. 37,240

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0013002
Application Number

출 원 년 월 일 : 2003년 03월 03일
Date of Application MAR 03, 2003

출 원 인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2003 년 04 월 01 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】 서지사항 보정서
【수신처】 특허청장
【제출일자】 2003.03.20
【제출인】
【명칭】 삼성전자 주식회사
【출원인코드】 1-1998-104271-3
【사건과의 관계】 출원인
【대리인】
【성명】 이영필
【대리인코드】 9-1998-000334-6
【포괄위임등록번호】 2003-003435-0
【대리인】
【성명】 이해영
【대리인코드】 9-1999-000227-4
【포괄위임등록번호】 2003-003436-7
【사건의 표시】
【출원번호】 10-2003-0013002
【출원일자】 2003.03.03
【발명의 명칭】 메타데이터 관리 방법
【제출원인】
【접수번호】 1-1-03-0074088-60
【접수일자】 2003.03.03
【보정할 서류】 특허출원서
【보정할 사항】
【보정대상항목】 우선권주장
【보정방법】 정정
【보정내용】
【우선권주장】
【출원국명】 US
【출원종류】 특허

【출원번호】 60/418,160
【출원일자】 2002.10.15
【증명서류】 미첨부
【우선권주장】
【출원국명】 US
【출원종류】 특허
【출원번호】 60/425,259
【출원일자】 2002.11.12
【증명서류】 미첨부

【취지】 특허법시행규칙 제13조·실용신안법시행규칙 제8조의 규정에의하여 위와 같 이 제출합니다. 대리인
이영필 (인) 대리인
이해영 (인)

【수수료】
【보정료】 0 원
【기타 수수료】 원
【합계】 0 원

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0004
【제출일자】	2003.03.03
【국제특허분류】	G06F
【발명의 명칭】	메타데이터 관리 방법
【발명의 영문명칭】	Method for managing metadata
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2003-003436-7
【발명자】	
【성명의 국문표기】	최양림
【성명의 영문표기】	CHOI,Yang Lim
【주민등록번호】	710120-1830615
【우편번호】	463-060
【주소】	경기도 성남시 분당구 이매동 124 한신아파트 210동 1509호
【국적】	KR
【우선권주장】	
【출원국명】	US
【출원종류】	특허
【출원번호】	00/000,000
【출원일자】	2002.10.14
【증명서류】	미첨부

【우선권주장】

【출원국명】	DE
【출원종류】	특허
【출원번호】	00/000,000
【출원일자】	2002.11.11
【증명서류】	미첨부
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 이영필 (인) 대리인 이해영 (인)

【수수료】

【기본출원료】	20	면	29,000 원
【가산출원료】	24	면	24,000 원
【우선권주장료】	2	건	43,000 원
【심사청구료】	0	항	0 원
【합계】	96,000 원		

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 메타데이터 전송 서버에서의 메타데이터 관리 방법에 관한 것으로서, 전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성하는 단계와, 생성된 프래그먼트 데이터들 중 소정의 프래그먼트 데이터를 선택하는 단계와, 상기 선택된 프래그먼트 데이터로부터 메타데이터 연관 정보를 생성하는 단계와, 상기 선택된 프래그먼트 데이터와 상기 생성된 메타데이터 연관 정보를 상기 메타데이터 연관 정보를 생성하기 위해 사용된 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보와 함께 전송하는 단계를 포함한다.

【대표도】

도 6

【명세서】

【발명의 명칭】

메타데이터 관리 방법{Method for managing metadata }

【도면의 간단한 설명】

도 1은 메타데이터 인증 레벨을 설명하기 위한 블록도

도 2는 물리 계층에서의 메타데이터 전달 방식을 설명하기 위한 도면

도 3은 단방향 채널에서의 메타데이터 콘테이너 레벨 인증에 사용되는 메타데이터 콘테이너의 포맷을 도시하는 도면

도 4는 양방향 채널에서의 메타데이터 콘테이너 레벨 인증에 사용되는 SOAP 메시지를 도시하는 도면

도 5는 메타데이터의 인덱싱 정보를 이용한 메타데이터 분류 방법을 설명하기 위한 도면

도 6은 본 발명의 일 실시예에 따른 메타데이터 전송 서버에서의 메타데이터 관리 방법을 설명하기 위한 흐름도

도 7은 본 발명의 일 실시예에 따른 메타데이터 클라이언트에서의 메타데이터 관리 방법을 설명하기 위한 흐름도

도 8은 본 발명의 일 실시예에 따른 메타데이터 전송 서버에서의 메타데이터 관리 방법을 설명하기 위한 흐름도

도 9은 본 발명의 일 실시예에 따른 메타데이터 클라이언트에서의 메타데이터 관리 방법을 설명하기 위한 흐름도

도 10은 단방향 채널의 경우의 데이터 콘테이너 포맷을 도시하는 도면

도 11은 양방향 채널의 경우의 SOAP 메시지를 도시하는 도면

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<12> 본 발명은 메타데이터 전송 서버 및 클라이언트에서의 메타데이터 관리 방법에 관한 것으로서, 특히 메타데이터가 전송 서버로부터 전송된 후, 클라이언트 장치에 의해 수신될 때까지의 메타데이터의 메시지 소스, 메시지 강건성, 및 기밀성의 인증과 관련된 메타데이터 관리 방법에 관한 것이다.

<13> 멀티미디어 시스템에서, 멀티미디어 콘텐트 및 그와 연관된 메타데이터는 서비스 제공자(service provider)로부터 클라이언트 장치로 제공된다. 이러한 멀티 시스템의 예는 데이터가 서버로부터 클라이언트로 전송되는 브로드캐스팅 시스템 또는 서버 및 클라이언트가 서로 상호작용하는 비디오 온 디맨드(video on demand: 팽)와 같은 서비스 방식이다.

<14> 전송되는 메타데이터는 클라이언트 장치에 의해 다양한 방식으로 사용된다. 메타데이터의 한 가지 사용예는 클라이언트 장치가 재생, 기록, 전송 등과 같은 동작을 수행하고자 하는 멀티미디어 콘텐트를 선택하는 것이 있다.

<15> 한편, 최근 방송 시스템에 있어서의, 클라이언트 장치에서 사용되는 메타데이터에 포함되는 정보는 점점 풍부해지고 복잡해지며, 그 보안의 중요성도 증대되고 있다. 따라서, 수신된 메타데이터의 경우 메타데이터가 생성된 후, 전송 서버

로부터 전송되어, 클라이언트에 의해 수신 될 때까지 소스 인증과 강건성 및 기밀성이 유지되었는지에 대한 인증의 필요성이 보다 증대하고 있지만, 이러한 메타 데이터의 인증을 효과적으로 수행하기 위한 메타데이터 관리 방법이 존재하지 않았다.

【발명이 이루고자 하는 기술적 과제】

<16> 본 발명은 상기와 같은 필요성을 만족하기 위해, 전송되는 메타데이터의 인증이 효과적으로 수행되도록 하기 위한 메타데이터 전송 서버에서의 메타데이터 관리 방법을 제공하는 것을 목적으로 한다.

<17> 또한, 본 발명은 수신된 메타데이터의 인증이 효과적으로 이루어지도록 하는 클라이언트에서의 메타데이터 관리 방법을 제공하는 것을 목적으로 한다.

【발명의 구성 및 작용】

<18> 상기 목적은 메타데이터 전송 서버에서의 메타데이터 관리 방법에 있어서, 전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성하는 단계와, 생성된 프래그먼트 데이터들 중 소정의 프래그먼트 데이터를 선택하는 단계와, 상기 선택된 프래그먼트 데이터로부터 메타데이터 연관 정보를 생성하는 단계와, 상기 선택된 프래그먼트 데이터와 상기 생성된 메타데이터 연관 정보를 상기 메타데이터의 연관 정보를 생성하기 위해 사용된 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보와 함께 전송하는 단계를 포함하는 관리 방법에 의해 달성된다.

<19> 또한, 상기 목적은 메타데이터를 수신하는 클라이언트에서의 메타데이터 관리 방법에 있어서, 상기 수신된 메타데이터 중 소정의 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보 및 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보를 판독하

는 단계와, 상기 판독된 프래그먼트 데이터와 대응하는 프래그먼트 데이터 포맷 정보를 사용하여 메타데이터 연관 정보를 생성하는 단계와, 상기 생성된 메타데이터 연관 정보와 상기 수신된 메타데이터 연관 정보를 비교하여 상기 수신된 메타데이터의 인증 여부를 결정하는 단계를 포함하는 관리 방법에 의해 달성된다.

<20> 또한, 상기 목적은 메타데이터를 수신하는 클라이언트에서의 메타데이터 관리 방법에 있어서, 상기 수신된 메타데이터 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보, 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보, 메타데이터 인증 서명 정보, 및 암호화된 제1 암호화키를 수신하는 단계와, 상기 수신된 메타데이터 프래그먼트 데이터와 대응하는 데이터 포맷 정보를 사용하여 메타데이터 연관 정보를 생성하는 단계와, 상기 수신된 암호화된 제1 암호화키를 상기 클라이언트에 저장된 제2 암호화키를 사용하여 복호화하는 단계와, 상기 생성된 메타데이터 연관 정보와 상기 복호화된 제1 암호화키를 사용하여 메타데이터 인증 서명 정보를 생성하는 단계와, 상기 생성된 메타데이터 인증 서명 정보와 상기 수신된 메타데이터 인증 서명 정보를 비교하여, 상기 수신된 메타데이터의 인증 여부를 결정하는 단계를 포함하는 관리 방법에 의해 달성된다.

<21> 본 발명에 따른 메타데이터 인증 방법은 메타데이터가 서버로부터 클라이언트로 전송되는 동안 변경되었는지 여부를 확인하고, 해당 메타데이터가 어떤 서비스 제공자 또는 메타데이터 콘텐트 제공자로부터 전송되는지를 효과적으로 인증하기 위한 전송 서버 및 클라이언트 장치에서의 메타데이터 관리 방법에 관한 것이다.

<22> 수신자가 메타데이터를 수신하는 경우, 수신된 메타데이터의 인증(authentication)이 이루어져야 한다. 여기에는 전송 레벨에서의 인증 및 소스 레벨에서의 인증으로 크게 나눌 수 있다.

<23> 전송 레벨에서의 인증은 전송 레벨에서의 메시지 소스(message source), 메시지 강건성(message integrity), 및 메시지 기밀성(message confidentiality)에 대한 인증을 수행한다.

<24> 전송 레벨에서의 메시지 소스의 인증은 메시지, 즉 메타데이터 콘텐트를 생성한 소스에 대한 인증이 아니라, 메시지를 전송하는 소스를 인증하는 것이다. 예를 들어, 도 1에 도시된 바와 같이, 메타데이터 콘텐트 제공자(metadata content provider)(120)가 존재하고, SK 텔레콤과 같은 서비스 제공자(140)가 별도로 존재하는 경우, 클라이언트(160)가 수신하는 메타데이터 A에 대한 전송 레벨에서의 메시지 소스 인증은 서비스 제공자(140)가 메타데이터 A를 클라이언트(160)로 전송했다는 것에 대한 인증을 의미한다.

<25> 또한, 전송 레벨에서의 메시지 강건성에 대한 인증은 서비스 제공자(140)로부터 클라이언트(160)로 전송된 메타데이터 A가 전송 도중 어떠한 변경이 없었다는 것을 인증하는 것이다.

<26> 또한, 전송 레벨에서의 메시지 기밀성에 대한 인증은 서비스 제공자(140)로부터 클라이언트(160)로 전송된 메타데이터 A가 전송 도중 공개되지 않는다는 것을 인증하는 것이다. 이러한 전송 레벨에서의 인증은 TCP/IP 프로토콜에서의 SSL/TLS 알고리즘, IEEE 1394 프로토콜에서의 DTCP 알고리즘, 및 DVI 프로토콜에서의 HDCP 알고리즘에 의해 이루어진다.

<27> 소스 레벨에서의 인증은 소스 레벨(source level)에서 메시지 소스, 메시지 강건성, 및 메시지 기밀성에 대한 인증을 수행한다.

<28> 소스 레벨에서의 메시지 소스 인증은 메시지, 즉 메타데이터 콘텐트를 생성한 소스를 인증하는 것이다. 예를 들어, 도 1에 도시된 바와 같이, 메타데이터 콘텐트 제공자(120)가 존재하고, SK 텔레콤과 같은 서비스 제공자(140)가 별도로 존재하는 경우, 클라이언트(160)가 수신하는 메타데이터 A에 대한 소스 레벨에서의 메시지 소스 인증은 메타데이터 콘텐트 제공자(120)가 해당 메타데이터 A를 클라이언트(160)로 전송했다는 것에 대한 인증을 의미한다.

<29> 또한, 소스 레벨에서의 메시지 강건성에 대한 인증은 메타데이터 콘텐트 제공자(120)로부터 클라이언트(160)로 전송된 메타데이터 A가 전송 도중 어떠한 변경이 없었다는 것을 인증하는 것이다.

<30> 또한, 소스 레벨에서의 메시지 기밀성에 대한 인증은 메타데이터 콘텐트 제공자(120)로부터 클라이언트(160)로 전송된 메타데이터 B가 전송 도중 공개되지 않는 다는 것을 인증하는 것이다.

<31> 따라서, 소스 레벨에서의 메타데이터 인증이 이루어지는 경우에는 전송 레벨에서의 메타데이터 인증이 별도로 이루어질 필요가 없다.

<32> 도 2(a), (b), 및 (c)는 각각의 레벨에 있어서의 물리 계층에서의 메타데이터 전달 방식을 설명하기 위한 도면이다.

<33> 도 2(a)은 전송 레벨 인증시 사용되는 전송 패킷을 도시하고 있다. 전송 레벨에서의 인증은 도 2(a)에 도시된 각각의 전송 패킷들에 대해 이루어지며, 이들 전송 패킷은 이진 XML 타입이다.

<34> 도 2(c)는 소스 레벨 인증시 사용되는 메타데이터를 도시한다. 도 2(c)의 메타데이터는 텍스트 XML 타입이다.

<35> 도 2(b)는 본 발명의 일 실시예를 설명하기 위한, 메타데이터 콘테이너 레벨(metadata container level) 인증시 사용되는 메타데이터 콘테이너를 도시한다. 이들 메타데이터 콘테이너에는 메타데이터의 의미 있는 세그먼트 단위로 삽입된다. 이들 메타데이터 콘테이너의 예는 도 3 및 도 4에 도시되어 있다.

<36> 도 3은 단방향 채널에서의 메타데이터 콘테이너 레벨 인증에 사용되는 메타데이터 콘테이너의 포맷을 도시하는 도면이다.

<37> 도 3에 도시된 바와 같이, 메타데이터 콘테이너는 헤더, 프래그먼트 데이터(fragment data), 및 메타데이터 인증 정보 등을 포함하며, 콘테이너 헤더에는 메타데이터 콘테이너 레벨에서의 인증을 위한 제어 정보가 포함되어 있다.

<38> 제어 정보에는 제1 제어 정보 F_1, 제2 제어 정보 F_2, 제3 제어 정보 F_3, 제4 제어 정보 F_4, 및 제5 제어 정보 F_5가 있으며, 이들 제어 정보들은 하나의 신호 또는 플래그로 이루어진다.

<39> 제1 제어 정보 F_1은 해당 콘테이너가 운반하는 메타데이터의 프래그먼트 데이터에 메타데이터 콘테이너 레벨의 인증을 위한 방법이 적용되었는지 여부를 나타내는 인증 플래그이다. 메타데이터 콘테이너 인증을 위한 방법으로는 미디어 액세스 제어(media

authentication code: MAC) 또는 전자 서명 알고리즘(digital signature algorithm: DSA) 등이 있다.

<40> 제2 제어 정보 F_2는 메타데이터 콘테이너 레벨에서의 인증 정보를 생성하기 위해 사용되는 특정 알고리즘을 나타내기 위한 정보이다. 제2 제어 정보 F_2는 하나의 이진 코드 집합을 사용하여 표현될 수 있고, 이러한 이진 코드와 특정 알고리즘들 간의 관계는 미리 정의되어 있고, 서비스를 제공하는 서버 및 메타데이터 콘테이너를 수신하는 클라이언트에 미리 알려져 있다.

<41> 제3 제어 정보 F_3은 F_2에 의해 특정되는 알고리즘이 메타데이터 콘테이너에 함께 포함된 프래그먼트 데이터에 어떠한 방식으로 구체적으로 사용되는지를 나타내기 위한 데이터 포맷 정보이다. 이는, 메타데이터 콘테이너 레벨 인증 알고리즘이 적용된 프래그먼트 데이터는 텍스트 형태로부터 변환된 이진 XML 형태일 수 있고, 또는 원래의 텍스트 XML 일 수도 있기 때문이다.

<42> 제3 제어 정보 F_3가 필요한 이유는, 본 발명에 따른 메타데이터의 인증 정보의 생성은 메타데이터를 해시 함수에 입력하여 얻은 출력 값, 즉 해시 값(hash value)을 이용하여 이루어지기 때문에, 텍스트 XML 데이터의 인증 정보는 대응 이진 XML 데이터의 인증 서명 정보와 관련성이 없기 때문이다. 즉, 클라이언트 장치에서 수신된 메타데이터 콘테이너에 포함된 메타데이터와 해시 값으로부터 인증 서명의 유효 여부를 판단하기 위해서는 해시 값을 계산하기 위해 사용된 메타데이터의 포맷을 알고 있어야 하기 때문이다.

<43> 제4 제어 정보 F_4는 메타데이터 인증과 관련된 암호화 키 정보를 의미한다. 암호화 키 정보는 메타데이터와 함께 메타데이터 콘테이너에 삽입되어 전송 서버로부터 클라

이언트 장치로 전송되거나, 선택적으로 별도의 보안 채널을 통해 전송 서버로부터 클라이언트 장치로 전송된다.

<44> 제5 제어 정보 F_5는 적용된 인증 레벨을 표시하는 인증 레벨 플래그로서, 적용된 인증 레벨이 전송 레벨인지 또는 전송 레벨인지 여부를 표시한다. 예를 들어, F_5가 '0'으로 설정된 경우에는 인증 레벨이 전송 레벨임을 표시하고, F_5가 '1'로 설정된 경우에는 인증 레벨이 소스 레벨임을 표시한다. 이러한 인증 레벨 플래그를 사용함으로써, 클라이언트 장치의 애플리케이션은 전송된 메타데이터의 인증 레벨이 전송 레벨인지 또는 소스 레벨인지에 따라 메타데이터의 신뢰 정도를 판단하여 전송된 메타데이터의 사용 여부를 결정할 수 있게 된다.

<45> 또한, 메타데이터 콘테이너는 프래그먼트 데이터 저장 영역을 포함하며, 상기 프래그먼트 데이터 저장 영역에는 적어도 하나 이상의 프래그먼트 데이터가 삽입된다. 본 실시예에 따른 콘테이너에는 메타데이터의 각 의미 세그먼트 단위, 예를 들어 하나의 프로그램에 대한 프로그램 정보와 같은 프래그먼트 데이터가 삽입된다. 하지만, 선택적으로 임의의 단위의 메타데이터를 운반하는 경우에도 적용될 수 있다. 또한, 연관된 메타데이터는 일련의 콘테이너에 의해 서비스 제공자로부터 클라이언트 장치로 전송된다. 또한, 하나의 메타데이터 콘테이너는 하나의 메타데이터 프래그먼트 또는 다수의 프래그먼트를 포함한다. 예를 들어, 하나의 메타데이터의 프래그먼트 데이터는 XML 트리 구조로 표현된 전체 메타데이터 중 하나의 부-트리(sub-tree)이다.

<46> 또한, 메타데이터 콘테이너 레벨 인증 정보에는 메타데이터 다이저스트 정보와 인증 서명 정보가 있다.

<47> 메타데이터 다이저스트 정보는 프래그먼트 데이터 저장 영역에 저장된 프래그먼트 데이터들 중 하나를, 제2 제어 정보 F_2에 의해 특정된 해시 함수와 같은 일방향 함수에 삽입하여 얻어진 결과값을 의미한다. 각각의 메타데이터 다이저스트 정보는 포인터를 사용하여 각각의 대응 프래그먼트 데이터와 연관되어 있다. 예를 들어, 제1 다이저스트 정보는 포인터에 의해 제1 프래그먼트 데이터와 연관되어 있다. 본 실시예에서는, 메타데이터 다이저스트 정보를 생성하기 위해, 해시 함수를 사용하였지만, 선택적으로 일방향 함수의 특성을 갖는 함수를 사용하여 메타데이터 다이저스트 정보를 구하는 것도 가능하다.

<48> 인증 서명 정보(authentication signature information)는 다이저스트 정보와 암호화 키 K가 결합된 값을 제2 제어 정보 F_2에 의해 특정되는 해시 함수와 같은 일방향 함수에 삽입하여 얻어진 결과 값을 의미한다. 메타데이터 다이저스트 정보와 마찬가지로, 각각의 메타데이터 인증 서명 정보는 포인터를 사용하여 각각의 대응 프래그먼트 데이터와 연관되어 있다. 예를 들어, 제1 인증 서명 정보는 포인터에 의해 제1 프래그먼트 데이터와 연관되어 있다. 본 실시예에서는, 인증 서명 정보를 생성하기 위해, 해시 함수를 사용하였지만, 선택적으로 일방향 함수의 특성을 갖는 함수를 사용하여 인증 서명 정보를 구하는 것도 가능하다.

<49> 도 4는 양방향 채널에서의 메타데이터 콘테이너 레벨 인증에 사용되는 SOAP 엔벨로프의 포맷을 도시하는 도면이다.

<50> 도 4에 도시된 바와 같이, 인증 관련 정보는 SOAP 헤더에 포함되어 있으며, 메타데이터 프래그먼트 데이터는 본문에 포함되어 있다.

<51> 인증 관련 정보 중 'Algorithm ID' 정보, 'SignatureValueBaseType' 정보, 및 'KeyInfo' 정보는 도 3의 제2 제어 정보 F_2, 제3 제어 정보 F_3, 및 제4 제어 정보 F_4에 대응한다. 'Digest' 정보 및 'Signature Value' 정보는 도 3의 메타데이터 다이저스트 정보 및 메타데이터 인증 서명 정보에 각각 대응한다. 'Authenticational Level' 정보는 메타데이터의 인증 레벨을 특정하기 위한 정보로서, 도 3의 제5 제어 정보 F_5 인증 레벨 플래그에 대응한다.

<52> 도 3 및 도 4에 도시된 바와 같이, 메타데이터를 의미 단위의 세그멘테이션 단위로 분할된 프래그먼트 데이터를 메타데이터 콘테이너에 삽입함으로써, 효율적인 암호화 관리 및 메타데이터 관리가 가능하게 된다.

<53> 예를 들어, 각각의 프래그먼트 데이터 단위로 인덱싱 정보를 부가함으로써, 도 5에 도시된 바와 같이 캐시(520)로 입력된 메타데이터들 중 인덱스 리스트 저장부(522)에 저장된 인덱스 리스트에 기초해서 선별된 메타데이터들만 저장 장치(540)에 저장하는 것이 가능하다. 또한, 도 4에 도시된 바와 같이, 메타데이터 프래그먼트 데이터가 프로그램 정보, 세그멘테이션 정보 등의 의미 있는 단위로 분할되기 때문에, 이를 각각의 프래그먼트 데이터를 선택적으로 암호화 하는 것도 가능하게 된다.

<54> 도 6는 도 3 및 도 4에 도시된 메타데이터 콘테이너를 사용한 메타데이터 콘테이너 레벨 인증 방법에 있어서, 도 1의 메타데이터 콘텐트 제공 서버(120) 또는 서비스 제공 서버(140)에서 수행되는 절차를 설명하는 흐름도이다.

<55> 단계 610에서는 전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성한다. 본 실시예에서 생성되는 프래그먼트 데이터는 메타데이

터에서 예를 들어 하나의 프로그램에 대한 프로그램 정보와 같은 의미를 갖는 세그먼트 단위이다.

<56> 단계 620에서는 생성된 프래그먼트 데이터들 중에서 소정의 프래그먼트 데이터를 선택한다.

<57> 단계 630에서 선택된 프래그먼트 데이터를 해시 함수, 예를 들어 SHA-1과 같은 보안 해시 알고리즘(secure hash algorithm)에 삽입하여 얻어진 결과값인 메타데이터 다이저스트 정보를 생성한다. 본 실시예에서는 메시지 다이저스트 정보를 생성하기 위해, 해시 함수를 사용하였다. 하지만, 선택적으로 해시 함수와 같은 일방향 함수(one way function)의 특성을 갖는 다른 함수를 사용하는 것도 가능하다.

<58> 단계 640에서는 선택된 프래그먼트 데이터와 생성된 메타데이터 다이저스트 정보 및 선택된 프래그먼트 데이터의 포맷이 이진 XML인지 또는 텍스트 XML인지를 나타내는 데이터 포맷 정보를 포함하는 메타데이터 콘테이너를 생성한 후, 이를 클라이언트로 전송한다.

<59> 선택된 프래그먼트 데이터 타입을 표시하는 이유는, 메타데이터의 프래그먼트 데이터가 동일한 경우에도, 단계 620에서의 메타데이터 다이저스트 정보 생성시 사용되는 프래그먼트 데이터의 타입에 따라서 메타데이터 다이저스트 정보가 달라지기 때문이다.

<60> 단계 640에서 생성되는 메타데이터 콘테이너의 예는 도 3 및 도 4에 도시되어 있다. 또한, 선택적으로 단계 640에서는 생성된 메타데이터 콘테이너의 인증 플래그를 설정하여 해당 메타데이터 콘테이너가 운반하는 메타데이터의 프래그먼트 데이터에 메타데이터 콘테이너 레벨의 인증 방법이 적용되었음을 나타낸다.

<61> 선택적으로, 메타데이터 콘테이너에 메타데이터 다이저스트 정보 생성을 위해 사용된 알고리즘 정보를 삽입한다. 예를 들어, 단계 520에서 메타데이터 다이저스트 정보를 생성하기 위해 해시 함수를 사용한 경우, 해시 함수가 인증 정보 생성 알고리즘을 사용되었음을 나타내는 알고리즘 정보를 삽입한다. 하지만, 알고리즘 정보가 전송 서버와 클라이언트 사이에 알려져 있는 경우에는, 알고리즘 정보는 메타데이터 콘테이너에 삽입되지 않는다.

<62> 또한, 프래그먼트 데이터 타입 정보와 함께 메타데이터 인증 레벨을 특정하는 플래그를 삽입하는 것도 가능하다. 메타데이타 인증 레벨을 특정하기 위한 플래그는 메타데이터 콘테이너를 사용한 메타데이터 인증이 전송 레벨에서 이루어지는지 또는 소스 레벨에서 이루어지는 지를 특정한다.

<63> 또한, 생성되는 메타데이터 콘테이너에 삽입되는 프래그먼트 데이터가 복수개인 경우, 메타데이터 콘테이너에는 이들 각각에 대해 계산된 메타데이터 다이저스트가 포함되고, 이들 각각의 프래그먼트 데이터와 메타데이터 다이저스트 간의 연관 관계를 나타내는 포인터 정보가 함께 포함된다.

<64> 또한, 생성되는 메타데이터 콘테이너에 삽입되는 프래그먼트 데이터가 복수개인 경우, 메타데이터 콘테이너에는 이들 각각의 프래그먼트 데이터들에 대한 인덱싱 정보가 함께 포함된다.

<65> 도 7은 도 3 및 도 4에 도시된 메타데이터 콘테이너를 사용한 메타데이터 콘테이너 레벨 인증 방법에 있어서, 도 1의 메타데이터 클라이언트 서버(160)에서 수행되는 절차를 설명하는 흐름도이다.

<66> 단계 710에서는 메타데이터 콘텐트 제공 서버(120) 또는 서비스 제공 서버(140)로부터 전송된 메타데이터 콘테이너를 수신한다.

<67> 단계 720에서는 수신된 메타데이터 콘테이너의 헤더에 포함된 제1 제어 정보 F_1, 즉 인증 플래그를 판독한다.

<68> 단계 730에서는 단계 730에서의 인증 플래그 판독 결과, 해당 메타데이터 콘테이너에 삽입된 프래그먼트 데이터에 대해 인증 방법이 적용된 것으로 판단된 경우 단계 740으로 진행하고, 인증 방법이 적용되지 않은 것으로 판단되는 경우에는 단계 742로 진행한다.

<69> 단계 740에서는 제2 제어 정보 F_2, 즉 인증 정보 생성을 위해 사용된 알고리즘을 인식하여 메타데이터 콘테이너에 삽입되어 있는 메타데이터 다이저스트 정보를 생성하기 위해 적용된 알고리즘을 판독한다. 본 실시예에서 사용된 인증 정보 생성 알고리즘은 해시 함수이다. 한편, 인증 정보 생성 알고리즘이 전송 서버와 클라이언트 간에 미리 특정된 경우에는 상기 알고리즘 판독 단계는 생략한다.

<70> 또한, 제3 제어 정보 F_3, 즉 메타데이터 포맷 정보를 인식하여, 메타데이터 콘테이너에 삽입된 메타데이터 다이저스트의 계산을 위해 사용된 프래그먼트 데이터의 포맷을 인식한다.

<71> 단계 742에서는 메타데이터 콘테이너 레벨의 인증 절차를 종료한다.

<72> 단계 750에서는 메타데이터에서 소정의 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보를 판독한다.

<73> 단계 760에서는 단계 740에서 판독된 프래그먼트 데이터와 데이터 포맷 정보에 기초해서, 단계 740에서 인식된 알고리즘, 예를 들어 해시 함수를 사용하여 메타데이터 다이저스트 정보를 생성한다.

<74> 단계 770에서는 단계 760에서 생성된 메타데이터 다이저스트 정보와 단계 750에서의 소정의 프래그먼트 데이터에 대응하는 메타데이터 다이저스트 정보를 비교하여, 전송된 메타데이터의 인증의 유효 여부를 결정한다.

<75> 또한, 선택적으로, 수신된 메타데이터 콘테이너에는 메타데이터 인증 레벨 플래그가 더 포함되어 있으며, 클라이언트 장치의 애플리케이션은 설정된 메타데이터 인증 레벨을 판독함으로써, 메타데이터의 인증이 전송 레벨로 설정되었는지 또는 소스 레벨로 설정되었는지를 알수 있고, 따라서 전송된 메타데이터의 신뢰 정도를 판단하여, 전송된 메타데이터의 사용 여부를 결정하는 것이 가능하게 된다.

<76> 도 8은 도 3 및 도 4에 도시된 메타데이터 콘테이너를 사용한 메타데이터 콘테이너 레벨 인증 방법에 있어서, 도 1의 메타데이터 콘텐트 제공 서버(120) 또는 서비스 제공 서버(140)에서 수행되는 절차를 설명하는 흐름도이다.

<77> 단계 810에서는 전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성한다. 본 실시예에서 생성되는 프래그먼트 단위는 메타데이터에서 예를 들어 하나의 프로그램에 대한 프로그램 정보와 같은 의미를 갖는 세그먼트 단위이다.

<78> 단계 820에서는 생성된 프래그먼트 데이터들 중에서 소정의 프래그먼트 데이터를 선택한다.

<79> 단계 830에서 선택된 프래그먼트 데이터를 해시 함수에 삽입하여 얻어진 결과값인 메타데이터 다이저스트 정보를 생성한다. 본 실시예에서는 메시지 다이저스트 정보를 생성하기 위해, 해시 함수를 사용하였다. 하지만, 선택적으로 해시 함수와 같은 일방향 함수의 특성을 갖는 다른 함수를 사용하는 것도 가능하다.

<80> 단계 840에서는 단계 830에서 생성된 메타데이터 다이저스트 정보와 암호화 키 K를 해시 함수에 입력하여 메타데이터 인증 서명(metadata authentication signature)을 생성한다. 사용된 암호화 키 K는 서비스 제공자에게 특유한 것이다. 본 실시예에서는 메시지 다이저스트 정보를 생성하기 위해, 해시 함수를 사용하였다. 하지만, 선택적으로 해시 함수와 같은 일방향 함수의 특성을 갖는 다른 함수를 사용하는 것도 가능하다. 메타데이터 인증 서명을 생성하기 위해 사용된 암호화 키는 또 다른 암호화 키 L을 사용하여 암호화 된다. 여기에서, 암호화 키 L을 사용하여 암호화된 암호화 키 K 값을 E(K)라고 한다. 계산된 암호화 키 E(K)는 메타데이터 콘테이너에 삽입되어 운반되거나, 또는 별도의 채널을 통해 클라이언트 장치로 전송된다. 또한, 암호화 키 L은 별도의 보안 채널을 사용하여 클라이언트 장치로 전송된다.

<81> 단계 850에서는 선택된 프래그먼트 데이터와 대응하는 메타데이터 다이저스트 정보, 메타데이터 인증 서명, 및 선택된 프래그먼트 데이터의 포맷 정보를 포함하는 메타데이터 콘테이너를 생성한 후, 이를 클라이언트로 전송한다.

<82> 단계 850에서 생성되는 메타데이터 콘테이너의 예는 도 3 및 도 4에 도시되어 있다. 또한, 선택적으로 단계 850에서는 생성된 메타데이터 콘테이너의 인증 플래그를 설정하여 해당 메타데이터 콘테이너가 운반하는 메타데이터의 프래그먼트 데이터에 메타데이터 콘테이너 레벨의 인증 방법이 적용되었음을 나타낸다.

<83> 선택적으로 생성된 메타데이터 콘테이너에 메타데이터 다이저스트 정보 생성을 위해 사용된 알고리즘 정보를 삽입한다.

<84> 또한, 선택된 프래그먼트 데이터의 포맷 정보는 메타데이터 다이저스트 정보 및 인증 정보를 생성하기 위해 사용된 프래그먼트 데이터의 포맷이 예를 들어 이진 XML 인지 또는 텍스트 XML인지 여부를 나타낸다.

<85> 또한, 생성되는 메타데이터 콘테이너에 삽입되는 프래그먼트 데이터가 복수개인 경우, 메타데이터 콘테이너에는 이들 각각에 대해 계산된 메타데이터 다이저스트 정보 및 인증 서명 정보가 포함되고, 이들 각각의 프래그먼트 데이터와 메타데이터 다이저스트 정보 및 메타데이터 인증 서명 정보 간의 연관 관계를 나타내는 포인터 정보가 함께 포함된다.

<86> 도 9는 도 3 및 도 4에 도시된 메타데이터 콘테이너를 사용한 메타데이터 콘테이너 레벨 인증 방법에 있어서, 도 1의 메타데이터 클라이언트 서버(160)에서 수행되는 절차를 설명하는 흐름도이다.

<87> 단계 910에서는 메타데이터 콘텐트 제공 서버(120) 또는 서비스 제공 서버(140)로부터 전송된 메타데이터 콘테이너를 수신한다.

<88> 단계 920에서는 수신된 메타데이터 콘테이너의 헤더에 포함된 제1 제어 정보 F_1, 즉 인증 플래그를 판독한다.

<89> 단계 930에서는 단계 920에서 인증 플래그 판독 결과, 해당 메타데이터 콘테이너에 삽입된 프래그먼트 데이터에 대해 인증 방법이 적용된 것으로 판단된 경우 단계 940으로

진행하고, 인증 방법이 적용되지 않은 것으로 판단되는 경우에는 단계 942으로 진행한다

.

<90> 단계 940에서는 제2 제어 정보 F_2, 즉 인증 정보 생성을 위해 사용된 알고리즘을 인식하여 메타데이터 콘테이너에 삽입되어 있는 메타데이터 다이저스트 정보 X를 생성하기 위해 적용된 알고리즘을 판독한다. 본 실시예에서 사용된 인증 정보 생성 알고리즘은 해시 함수이다. 인증 정보를 생성하기 위한 알고리즘이 전송 서버와 클라이언트 간에 이미 특정되어 있는 경우에는 상기 적용 알고리즘 판독 단계는 생략한다.

<91> 또한, 제3 제어 정보 F_3, 즉 프래그먼트 데이터의 포맷 정보를 인식하여, 메타데이터 콘테이너에 삽입된 메타데이터 다이저스트 정보의 계산을 위해 사용된 프래그먼트 데이터의 포맷을 인식한다.

<92> 단계 940에서는 메타데이터 콘테이너 레벨의 인증 절차를 종료한다.

<93> 단계 950에서는 메타데이터에서 소정의 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보를 판독한다.

<94> 단계 960에서는 판독된 프래그먼트 데이터와 데이터 포맷 정보에 기초해서, 단계 940에서 판독된 알고리즘, 예를 들어 해시 함수를 사용하여 메타데이터 다이저스트 정보를 생성한다.

<95> 단계 970에서는 클라이언트 장치에 저장된 암호화 키 L를 사용하여 암호화된 키 K를 복호화한다. 암호화 키 L은 메타데이터 전송 서버로부터 별도의 보안 채널을 통해 전송된 것이다.

<96> 단계 980에서는 단계 960에서 생성된 메타데이터 다이저스트 정보와 복호화된 키 K로부터 메타데이터 인증 서명 S를 생성한다.

<97> 단계 990에서는 단계 980에서 생성된 메타데이터 인증 서명 정보와 단계 950에서 판독된 인증 서명 정보를 비교하여, 전송된 메타데이터 인증 서명의 유효 여부를 결정한다.

<98> 또한, 선택적으로, 수신된 메타데이터 콘테이너에는 메타데이터 인증 레벨을 표시하는 인증 레벨 플래그가 더 포함되어 있으며, 클라이언트 장치의 애플리케이션을 메타데이터 인증 레벨을 판독하여, 인증 레벨의 타입에 따라 메타데이터 사용 여부를 결정한다.

<99> 또한, 별도의 이용 가능한 강건성 확인 방법이 존재한다. 이러한 예들 중의 하나는 공개키(public key)를 사용한 암호 작성법(cryptography)이다. 이 경우, 서비스 제공자는 비밀키(secret key) 및 공개키로 이루어진 한쌍의 키, 즉 (K_s, K_p)를 보유하고 있으며, K_s를 사용하여 메시지에 서명한다. 여기에서, K_s는 비밀키, K_p는 공개키를 의미한다.

<100> 클라이언트 장치는 신뢰할 만한 소스를 통해 공개키를 획득하는 것이 가능하다. 따라서, 클라이언트가 서명을 갖는 메타데이터 콘테이너를 수신하는 경우, 클라이언트 장치는 수신된 메타데이터 콘테이너가 전송되는 서비스 제공자를 확인하고, 확인된 서비스 제공자에 대응하는 공개키 K_p를 획득한다. 클라이언트는 수신된 서명이 유효한지 여부를 확인하기 위해 공개키를 사용한다.

<101> 이하에서는, 메타데이터의 보안을 유지하기 위한 메타데이터 인증 요건 및 방법에 대해 보다 구체적으로 설명한다.

<102> 메타데이터에 대한 적절한 보안을 유지하기 위해서는 메타데이터 액세스 및 이용에 대한 인증과, 메타데이터의 강건성 및 메타데이터의 기밀성 유지와, 메타데이터의 부분 집합 및 이진 포맷 및 텍스트 포맷에 대한 효율적인 보호가 필요하다.

<103> 즉, 애플리케이션에 의한 메타데이터 또는 메타데이터 일부에 대한 액세스 인증은 적절한 인증 규칙(authorization rule)을 따라야 한다. 이러한 인증 절차는 애플리케이션 단위 또는 애플리케이션 및 메타데이터 단위로 이루어진다.

<104> 또한, 메타데이터 전체 또는 메타데이터 일부의 액세스를 통한, 사용예에는 시청(view), 변형(modify), 및 복사(copy) 등이 있다. 시청은 액세스를 얻는 것과 거의 동일한 가장 간단한 사용 예이다. 메타데이터의 변형 또는 로컬 복사를 제어하기 위해서는 메타데이터 파일 관리 시스템을 필요로 한다. 또한, 원격 애플리케이션으로 메타데이터를 복사하는 것, 예를 들어 클라이언트가 메타데이터를 서비스 제공자로 전송하는 것은 메타데이터 요청에 대한 인증과, 보안 인증 채널을 통해 요청된 데이터 및 소스 인증 정보를 전송하는 것을 필요로 한다.

<105> 또한, 메타데이터에 대한 보안을 유지하기 위해서는 메타데이터의 기밀성 유지가 필요하다. 메타데이터의 인도 및 저장 동안, 메타데이터 중 일부분은 높은 가치 또는 프라이버시와 관련된 데이터를 포함하고 있는 등의 여러 가지 이유로 암호화될 필요가 있다. 이를 위해, 전송 레벨, 즉 전송되는 동안의 기밀성은 메타데이터의 전송 유닛 또는 콘테이너(container)를 암호화함으로써 유지될 수 있다. 또한, 소스 레벨에서의 암호화는 전송 및 저장 레벨에서의 기밀성과 관련된 문제를 모두 해결한다.

<106> 아래에서는, 조건적인 액세스 시스템과 관련된 단방향(uni_directional) 환경 및 양방향 채널(TLS)에서의 메타데이터 보안에 대해 설명한다.

<107> 조건적인 액세스 시스템과 관련된 단방향 환경, 즉 브로드캐스트 환경의 예는 지상파 방송 (ATSC,DVB), 위성 방송 (Direct TV), 케이블 TV, 및 IP-멀티캐스트등이 있다. 이들은 트랜잭션(transaction)과 같은 정보의 교환을 위해 사용되는 별도의 복귀 채널(return channel)을 제외하고는 단방향(uni-directional) 채널이다. 아래의 기능은 이러한 환경에서 지원된다.

<108> 하드웨어 장치를 갖춘 가입된 수신기와 송신기 간의 인증은 자동적으로 이루어진다. 또한, 수신기 및 송신기는 메인 브로드캐스트 채널과는 별도의 채널을 이용하여 커먼 시크리트(common secret)를 공유한다. 여기에서, 커먼 시크리트는 수신기와 송신기만이 공유하고 있는 별도의 코드를 의미한다. 패킷 페이로드는 암호화되어 전송된 후, 상기 커먼 시크리트를 사용하여 해독하거나 또는 상기 커먼 시크리트를 사용하여 해독된 키를 사용하여 해독된다.

<109> 양방향 채널 환경하에서는, 핸드쉐이크 프로토콜(handshake protocol)을 사용하며, 서버 또는 클라이언트는 제3의 증명서 인증 기관에 의해 발행된 증명서를 교환할 수 있도록 인증된다. 커먼 시크리트는 클라이언트와 서버 간에 공유되고, 이 후 세션 키가 생성된다. 패킷 페이로드는 세션 키를 사용하여 암호화된 후 전송되고, 이 후 동일한 세션키를 사용하여 복호화된다. 소스 인증은 전자 서명 알고리즘(digital signature algorithm: DSA) 또는 미디어 액세스 제어(media access control: MAC) 등과 같은 알고리즘을 통해 이루어질 수 있다.

<110> 또한, 양방향 채널 환경하에서는 클라이언트/서버의 인증은 신뢰할 수 있는 제3의 기관에 의한 증명서의 인증 및 교환을 통해 이루어지며, 수신된 데이터의 인증 및 전송 동안의 기밀 유지는 패킷 페이로드의 암호화 및 메시지 인증에 의해 이루어진다.

<111> 이와 같이, 메타데이터 전송과 관련된 보안을 만족하기 위해서는, 송신기와 수신기에 있어서 상호간의 인증이 이루어지고, 데이터 인증 및 데이터를 암호화하여 전송하는 것이 가능하도록, 커먼 시크리트가 안전하게 공유되어야 한다.

<112> 이하에서는, 전송 레벨 및 소스 레벨에서의 메타데이터에 대한 보호 방식에 대해 기술한다.

<113> 전송 중에 있어서의 메타 데이터 보호와 관련하여, 송신기 및 수신기의 인증은 전송 레벨에서 이루어지고, 메타데이터 인증 및 기밀 유지는 방송 시스템 레벨에서 이루어진다.

<114> 예를 들어, 단방향 채널의 경우에는, 각각의 SOAP 응답 (헤더 + 본문)이 보호 단위로서 사용될 수 있다. 단방향 경우의 이러한 방법의 예는 도 10에 도시되어 있다. 또한, 양방향 채널의 경우에는, 데이터 서명 정보는 SOAP 응답을 사용하여 전송될 수 있다. 이 경우, 도 11에 도시된 바와 같이, 서명 정보는 SOAP 헤더에 포함되어 있으며, 데이터 정보는 본문에 포함되어 있다. 본문의 데이터 부분들은 암호화될 수 있다.

<115> 또한, 소스 레벨에서의 메타데이터 보호와 관련하여, 아래에서는 방송 장치 내에서의 메타데이터의 강건성 및 기밀 유지 및 메타데이터 액세스 및 사용 제어에 대하여 기술한다.

<116> 방송 장치에서의 메타데이터의 강건성 및 기밀 유지는 인증 서명을 메타데이터에 연관시키고, 이를 암호화함으로써 가능하다. 또한, 메타데이터의 모든 부분이 암호화되거나 강건성이 유지되어야 할 필요가 없다는 점을 고려하여, 포인터를 사용하여, 암호화 또는 인증 절차가 수행된 메타데이터의 특정 부분을 나타내는 것이 필요하다. 이러한 동작은 권리 관리 보호(right management protection: RMP) 시스템에 의해 이러한 포인터가 유지되는 소스 레벨에서 수행될 수 있다. 소스 레벨 서명을 사용함으로써, 메타데이터 소스는 실질적으로 인증될 수 있다. 물론, 메타데이터는 이런 정보, 즉 소스의 인증 서명을 미리 포함하고 있어야 한다.

<117> 또한, 메타데이터의 액세스 및 사용 제어를 위해서는 표준 액세스 및 사용 권리 기술(standard access and usage right description) 및 이에 대한 강제가 필요하다. 이와 관련된 표준 기술(standard description)은 XML 스키마의 하나의 형태를 취하거나 또는 의미상으로 명확한 의미를 갖는 하나의 집합의 데이터 요소의 형태를 취할 수 있다. 이러한 기술을 구성하는데 관련될 수 있는 기존의 도구들은 XrML, XACML, 및 SAML 등이 있다. 라이선스 기술(license description) 및 이용 규칙(usage rule)은 메타데이터로부터 분리될 수 있다.

<118> 또한, 선택적으로 메타데이터의 사용이 기술될 수 있는 부분적인 메타데이터의 수가 잠재적으로 많은 점을 고려하여, 액세스/사용 제어를 보다 단순히 하기 위해, 일단 하나의 애플리케이션의 액세스가 인증되는 경우, 해당 애플리케이션의 동작은 디폴트로 사용 규칙을 준수하는 것으로 간주하는 것도 가능하다.

<119> 또한, 이러한 문제와 연관된 것은 액세스/사용과 관련된 RMP 시스템에서의 애플리케이션 프로그램 인터페이스(application program interface: API)이다. 이는 액세스/

사용 제어 정보가 TVA RMP 시스템에 의해 관리되는 경우에 필요하다. 예를 들어, 애플리케이션 프로그램 인터페이스는 액세스를 요청 및 허가하고, 변형, 복사, 및 외부로 내보내는 (export) 동작을 수행한다.

<120> 상기에서 설명된 바와 같이, 구조적인 레벨에서 인증이 수행될 수 있는 몇가지 인증 종류가 있다.

<121> 첫 번째는 전송레벨에서 이루어지는 것이고, 두 번째는 단방향 채널에서의 콘테이너 또는 SOAP 메시지와 같은 양방향 채널에서 이루어지는 것이고, 세 번째는 소스 레벨에서 이루어지는 것이다.

<122> 소스 레벨에서의 인증은 인증이 이루어지는 메타데이터의 구체적인 부분들에 대해 포인터를 사용하여 인증 정보를 제공한다. SOAP 메시지 레벨 인증의 경우, 인증 정보는 SOAP 메시지의 본문에 포함된 메타데이터의 일부 또는 전부에 대한 포인터와 함께 헤더에 포함된다.

<123> 또한, 전송 중 단지 강건성 보증만이 요구되는 경우, 전송 레벨에서의 인증만으로 충분하다. 한편, 전송 독립성(transport independentness)이 필요한 경우, 콘테이너 레벨 또는 SOAP 메시지 레벨 인증이 이러한 요구 조건을 만족시킬 수 있다. 콘테이너 또는 SOAP 메시지 본문에 포함되는 메타데이터의 크기는 전송 패킷의 크기 보다 훨씬 크기 때문에, 이러한 전송 레벨 인증은 시스템에 주는 부하를 줄여준다. 물론, 보안 채널은 이러한 경우 유지될 필요가 없다.

<124> 메타데이터 소스의 인증을 위해서는, 콘테이너 및 SOAP 레벨 인증이 필요하다. 소스 인증이 가능하도록 하는 콘테이너의 신택스는 도 11에 도시된 바와 같다.

<125> 소스와 최종 수신지 사이의 많은 중간 노 드에 있어서도, 소스 인증이 이루어지도록 하기 위해서는, 각 중간 노드에서 소스 인증이 저장되어야 한다.

<126> 보다 구체적으로, 각각의 중간 노드에서 수신된 메타데이터는 이전 노드로부터 수신된 인증 정보를 사용하여 인증되고, 새로운 인증 정보가 생성되어, 다음 노드로 전달되거나, 또는 이전 노드로부터 전달된 메타데이터 및 인증 정보 전부가 다음 노드로 전달한다.

<127> 따라서, 몇 개의 중간 노드를 포함하는 소스 레벨 인증을 사용한 메타데이터 전달의 경우, 하나의 노드에서 이전 노드의 인증 정보를 사용하여 소스 레벨에서의 인증(authenticated) 이 후, 새로운 인증 정보가 생성되는지 여부를 나타내는 플래그 또는 신호가 인증 정보에 삽입될 수 있다. 이 플래그를 사용함으로써, 수신기는 소스 인증 정보의 유무에 따라 해당 메타데이터를 받아들일지 여부를 결정한다.

<128> 본 발명은 상술한 실시예에 한정되지 않으며, 본 발명의 사상내에서 당업자에 의한 변형이 가능함은 물론이다.

<129> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 하드디스크, 플로피디스크, 플래쉬 메모리, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브 (예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드로서 저장되고 실행될 수 있다.

【발명의 효과】

<130> 상술한 바와 같이 본 발명에 따른 메타데이터 관리 방법에 따르면, 메타데이터 콘테이너 레벨에서 메타데이터의 인증이 이루어지도록 함으로써, 채널 환경에 상관없이 전송 레벨에서의 인증이 가능하며, 또한 콘테이너에 인증을 위해 계산되는 메타데이터의 포맷을 나타내는 정보를 삽입하여 전송 레벨 및 소스 레벨에서의 인증 모두가 선택적으로 가능하도록 하며, 전송 레벨에서의 패킷에 비해, 메타데이터 콘테이너 레벨의 패킷의 크기가 비교적 크기 때문에, 전송되는 패킷의 수를 줄여 시스템 복잡도를 감소키는 것이 가능하다는 효과가 있다.

【특허청구범위】

【청구항 1】

메타데이터 전송 서버에서의 메타데이터 관리 방법에 있어서,

(a) 전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성하는 단계와,

(b) 생성된 프래그먼트 데이터들 중 소정의 프래그먼트 데이터를 선택하는 단계와,

(c) 상기 선택된 프래그먼트 데이터로부터 메타데이터 연관 정보를 생성하는 단계와,

(d) 상기 선택된 프래그먼트 데이터와 상기 생성된 메타데이터 연관 정보를 상기 메타데이터 연관 정보를 생성하기 위해 사용된 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보와 함께 전송하는 단계를 포함하는 것을 특징으로 하는 관리 방법.

【청구항 2】

제1항에 있어서, 상기 선택된 프래그먼트 데이터, 상기 생성된 메타데이터 연관 정보, 및 상기 프래그먼트 데이터의 포맷 정보는 하나의 메타데이터 콘테이너에 삽입되어 전송되는 것을 특징으로 하는 관리 방법.

【청구항 3】

제1항에 있어서, 상기 데이터 포맷 정보는 메타데이터 연관 정보 생성을 위해 사용된 프래그먼트 데이터가 이진 XML 포맷인지 또는 텍스트 XML 포맷인지를 나타내는 것을 특징으로 하는 방법.

【청구항 4】

제1항에 있어서, 상기 메타데이터 프래그먼트 데이터는 메타데이터의 의미 있는 세그먼트 단위인 것을 특징으로 하는 방법.

【청구항 5】

제2항에 있어서, 상기 메타데이터 콘테이너에는 메타데이터의 인증 레벨을 특정하는 인증 레벨 플래그가 더 삽입되는 것을 특징으로 하는 방법.

【청구항 6】

제1항에 있어서, 상기 메타데이터 연관 정보는 상기 선택된 프래그먼트 데이터를 일방향 함수(oneway function)에 입력하여 얻어진 메타데이터 다이저스트 정보인 것을 특징으로 하는 방법.

【청구항 7】

제6항에 있어서, 상기 일방향 함수는 해시 함수(hash function)인 것을 특징으로 하는 방법.

【청구항 8】

제1항에 있어서, 상기 생성된 메타데이터 연관 정보와 제1 암호화 키를 사용하여 메타데이터 인증 서명 정보를 생성하는 단계를 더 포함하며, 상기 생성된 메타데이터 인증 서명 정보를 상기 선택된 프래그먼트 데이터가 삽입된 메타데이터 콘테이너에 삽입하는 단계를 더 포함하는 것을 특징으로 하는 방법.

【청구항 9】

제8항에 있어서, 상기 메타데이터 인증 서명 정보는 상기 생성된 메타데이터 연관 정보와 제1 암호화 키를 일방향 함수에 입력하여 얻어진 결과값인 것을 특징으로 하는 방법.

【청구항 10】

제9항에 있어서, 상기 제1 암호화키를 제2 암호화키를 사용하여 암호화하는 단계와, 상기 암호화된 제1 암호화키를 상기 상기 선택된 프래그먼트 데이터가 삽입된 메타데이터 콘테이너에 삽입하는 단계를 더 포함하는 것을 특징으로 하는 방법.

【청구항 11】

제2항에 있어서, 상기 메타데이터 콘테이너에는 복수개의 프래그먼트 데이터 및 대응하는 복수개의 메타데이터 연관 정보가 삽입되며, 각각의 프래그먼트 데이터와 대응하는 메타데이터 연관 정보는 포인터 정보에 의해 연결되는 것을 특징으로 하는 방법.

【청구항 12】

제8항에 있어서, 상기 메타데이터 콘테이너에는 복수개의 프래그먼트 데이터와 대응하는 복수개의 메타데이터 연관 정보 및 복수개의 인증 서명 정보가 삽입되며, 각각의 프래그먼트 데이터와 대응하는 메타데이터 연관 정보 및 인증 서명 정보는 포인터 정보에 의해 연결되는 것을 특징으로 하는 방법.

【청구항 13】

메타데이터를 수신하는 클라이언트에서의 메타데이터 관리 방법에 있어서,

(a) 상기 수신된 메타데이터 중 소정의 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보 및 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보를 판독하는 단계와,

(b) 상기 판독된 프래그먼트 데이터와 대응하는 데이터 포맷 정보를 사용하여 메타데이터 연관 정보를 생성하는 단계와,

(c) 상기 (b) 단계에서 생성된 메타데이터 연관 정보와 상기 (a) 단계에서 수신된 메타데이터 연관 정보를 비교하여 상기 수신된 메타데이터의 인증 여부를 결정하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 14】

제13항에 있어서, 상기 수신된 프래그먼트 데이터와 대응하는 메타데이터 연관 정보 및 프래그먼트 데이터의 포맷 정보는 하나의 메타데이터 콘테이너에 포함되어 수신되는 것을 특징으로 하는 방법.

【청구항 15】

제13항에 있어서, 상기 데이터 포맷 정보는 메타데이터 연관 정보 생성을 위해 사용된 프래그먼트 데이터가 이진 XML 포맷 인지 텍스트 XML 포맷인지를 나타내는 것을 특징으로 하는 방법.

【청구항 16】

제13항에 있어서, 상기 프래그먼트 데이터는 메타데이터의 의미 있는 세그먼트 단위인 것을 특징으로 하는 방법.

【청구항 17】

제14항에 있어서, 상기 메타데이터 콘테이너에는 메타데이터 인증 레벨을 특정하는 인증 레벨 플래그가 포함되어 있는 것을 특징으로 하는 방법.

【청구항 18】

제13항에 있어서, 상기 메타데이터 연관 정보는 상기 선택된 프래그먼트 데이터를 일방향 함수에 입력하여 얻어진 메타데이터 다이저스트 정보인 것을 특징으로 하는 방법.

【청구항 19】

제18항에 있어서, 상기 일방향 함수는 해시 함수인 것을 특징으로 하는 방법.

【청구항 20】

제14항에 있어서, 상기 메타데이터 콘테이너에는 복수개의 프래그먼트 데이터 및 대응하는 연관 정보가 삽입되며, 각각의 프래그먼트 데이터와 대응하는 메타데이터 연관 정보는 포인터 정보에 의해 연결되는 것을 특징으로 하는 방법.

【청구항 21】

메타데이터를 수신하는 클라이언트에서의 메타데이터 관리 방법에 있어서,

(a) 상기 수신된 메타데이터 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보, 프래그먼트 데이터의 타입을 나타내는 데이터 포맷 정보, 메타데이터 인증 서명 정보, 및 암호화된 제1 암호화키를 수신하는 단계와,

(b) 상기 수신된 메타데이터 프래그먼트 데이터와 대응하는 데이터 포맷 정보를 사용하여 메타데이터 연관 정보를 생성하는 단계와,

(c) 상기 수신된 암호화된 제1 암호화키를 상기 클라이언트에 저장된 제2 암호화키를 사용하여 복호화하는 단계와,

(d) 상기 생성된 메타데이터 연관 정보와 상기 복호화된 제1 암호화키를 사용하여 메타데이터 인증 서명 정보를 생성하는 단계와,

(e) 상기 생성된 메타데이터 인증 서명 정보와 상기 수신된 메타데이터 인증 서명 정보를 비교하여, 상기 수신된 메타데이터의 인증 여부를 결정하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 22】

제21항에 있어서, 상기 메타데이터 연관 정보는 상기 선택된 프래그먼트 데이터를 일방향 함수에 입력하여 얻어진 메타데이터 다이저스트 정보인 것을 특징으로 하는 방법.

【청구항 23】

제22항에 있어서, 상기 일방향 함수는 해시 함수인 것을 특징으로 하는 방법.

【청구항 24】

제21항에 있어서, 상기 생성된 메타데이터 인증 서명 정보는 상기 생성된 메타데이터 연관 정보와 복호화된 제1 암호화 키를 일방향 함수에 입력하여 얻어진 결과값인 것을 특징으로 하는 방법.

【청구항 25】

제24항에 있어서, 상기 일방향 함수는 해시 함수 인 것을 특징으로 하는 방법.

【청구항 26】

제21항에 있어서, 상기 (a) 단계에서 수신된 프래그먼트 데이터와 이에 대응하는 메타데이터 연관 정보, 프래그먼트 데이터의 포맷 정보, 메타데이터 인증 서명 정보, 및 암호화된 제1 암호화키는 하나의 메타데이터 콘테이너에 포함되어 수신되는 것을 특징으로 하는 관리 방법.

【청구항 27】

제21항에 있어서, 상기 데이터 포맷 정보는 메타데이터 연관 정보 생성을 위해 사용된 프래그먼트 데이터가 이진 XML 포맷인지 또는 텍스트 XML 포맷인지를 나타내는 것을 특징으로 하는 방법.

【청구항 28】

제21항에 있어서, 상기 프래그먼트 데이터는 메타데이터의 의미 있는 세그먼트 단위인 것을 특징으로 하는 방법.

【청구항 29】

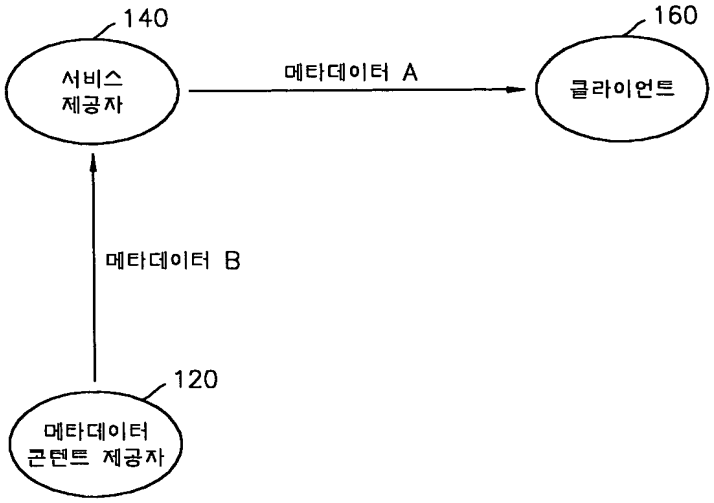
제26항에 있어서, 상기 메타데이터 콘테이너에는 메타데이터 인증 레벨을 특정하는 인증 레벨 플래그가 포함되어 있는 것을 특징으로 하는 방법.

【청구항 30】

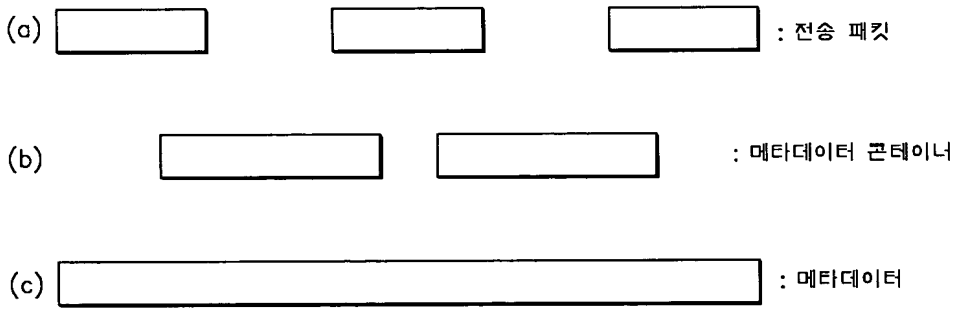
제26항에 있어서, 상기 메타데이터 콘테이너에는 복수개의 프래그먼트 데이터와 대응하는 복수개의 메타데이터 연관 정보 및 복수개의 인증 서명 정보가 삽입되며, 각각의 프래그먼트 데이터와 대응하는 메타데이터 연관 정보 및 인증 서명 정보는 포인터 정보에 의해 연결되는 것을 특징으로 하는 방법.

【도면】

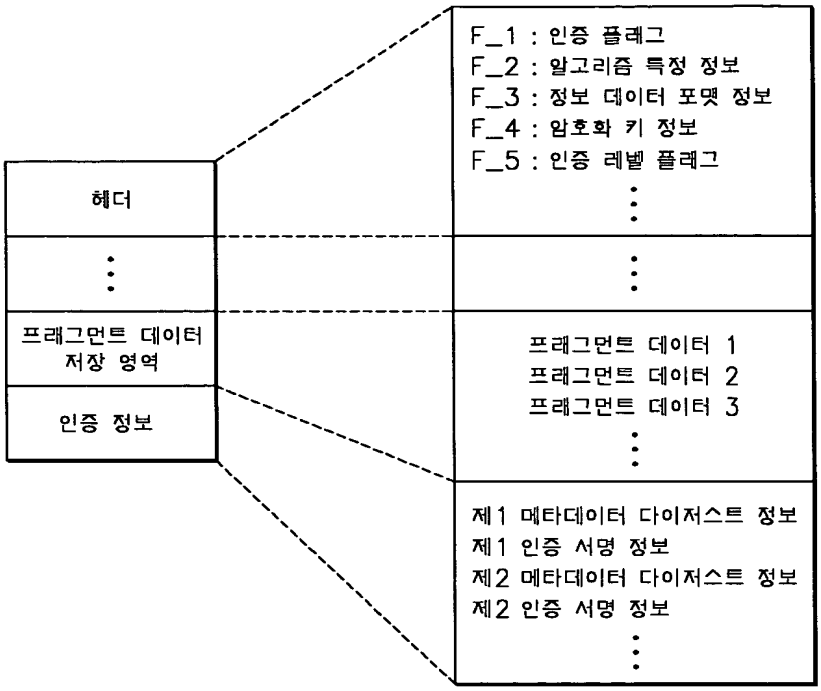
【도 1】


140
서비스
제공자
메타데이터 A
160
클라이언트
메타데이터 B
120
메타데이터
콘텐트 제공자

【도 2】


(a)
: 전송 패킷
(b)
: 메타데이터 콘테이너
(c)
: 메타데이터

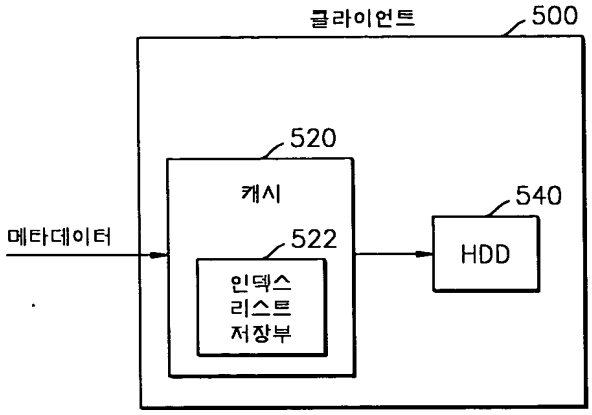
【도 3】



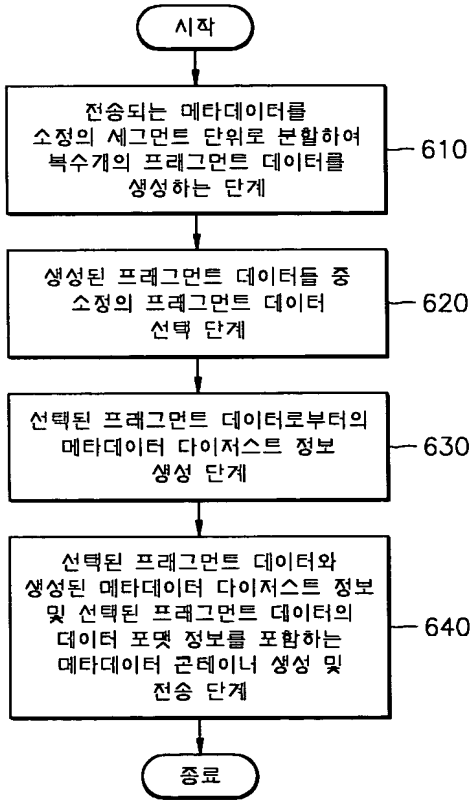
【도 4】

```
<SOAP:Envelope ...>
  <SOAP:Header>
      <Signature fragrefID = 1>
        <Algorithm ID=1>
        <Digest> ... <\Digest>
        <SignatureValue>  ...  </SignatureValue>
        <KeyInfo>  ... </KeyInfo>
        <SignatureValueBaseType>Text</SignatureValueBaseType>
        <AuthenticatioinLevel>Transport</AuthenticationLevel>
      </Signature>
  </SOAP:Header>
<SOAP:Body>
  <TVAmetadataFragment id=1>
      <ProgramInformation>...</ProgramInformation>
  </TVAmetadataFragment>
  <TVAmetadataFragment id=2>
      <SegmentInformation>...</SegmentInformation>
  </TVAmetadataFragment>
</SOAP:Body>
</SOAP:Envelope>
```

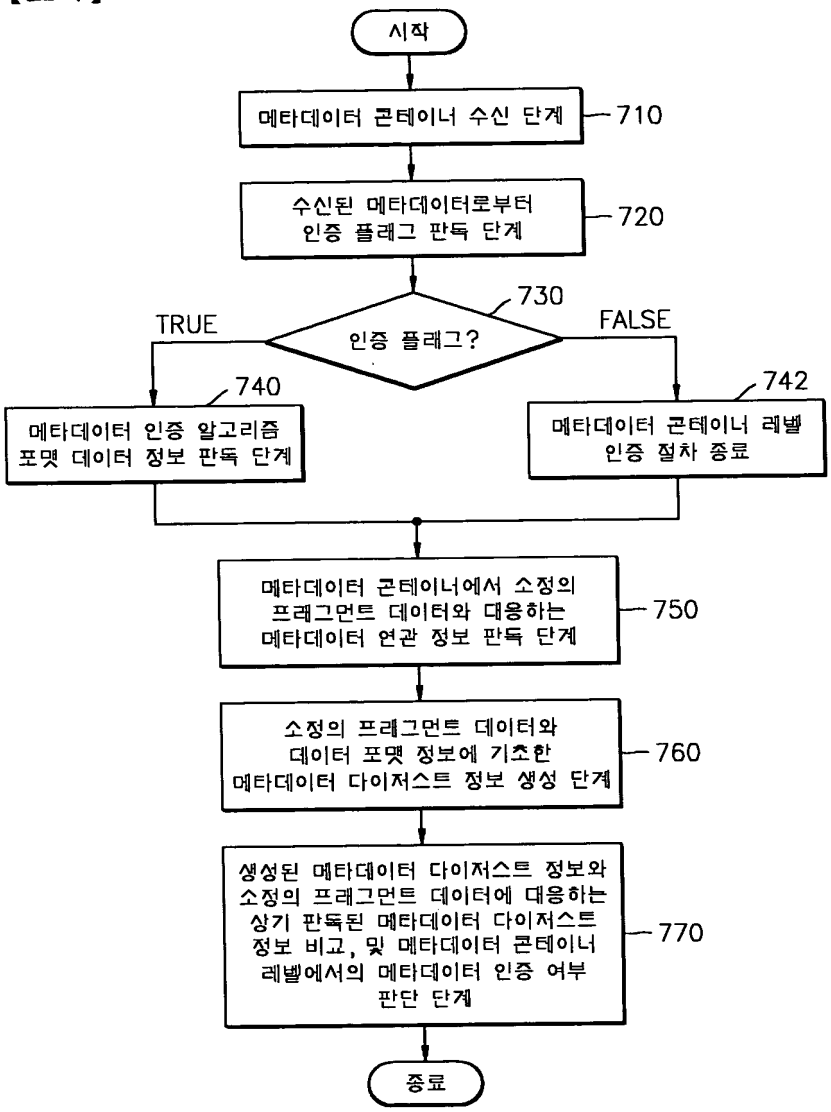
【도 5】


클라이언트
500
520
캐시
540
메타데이터
522
HDD
인덱스
리스트
저장부

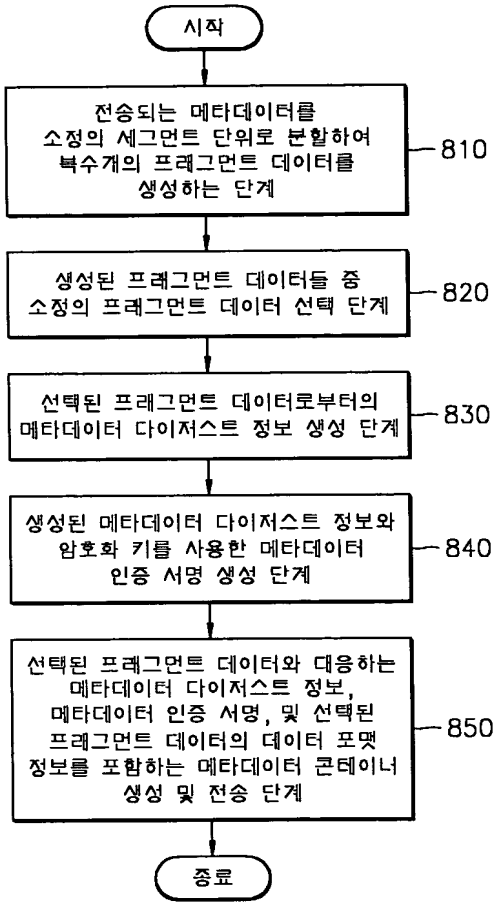
【도 6】


시작
전송되는 메타데이터를
소정의 세그먼트 단위로 분할하여
복수개의 프래그먼트 데이터를
생성하는 단계
610
생성된 프래그먼트 데이터들 중
소정의 프래그먼트 데이터
선택 단계
620
선택된 프래그먼트 데이터로부터의
메타데이터 다이저스트 정보
생성 단계
630
선택된 프래그먼트 데이터와
생성된 메타데이터 다이저스트 정보
및 선택된 프래그먼트 데이터의
데이터 포맷 정보를 포함하는
메타데이터 콘테이너 생성 및
전송 단계
640
종료

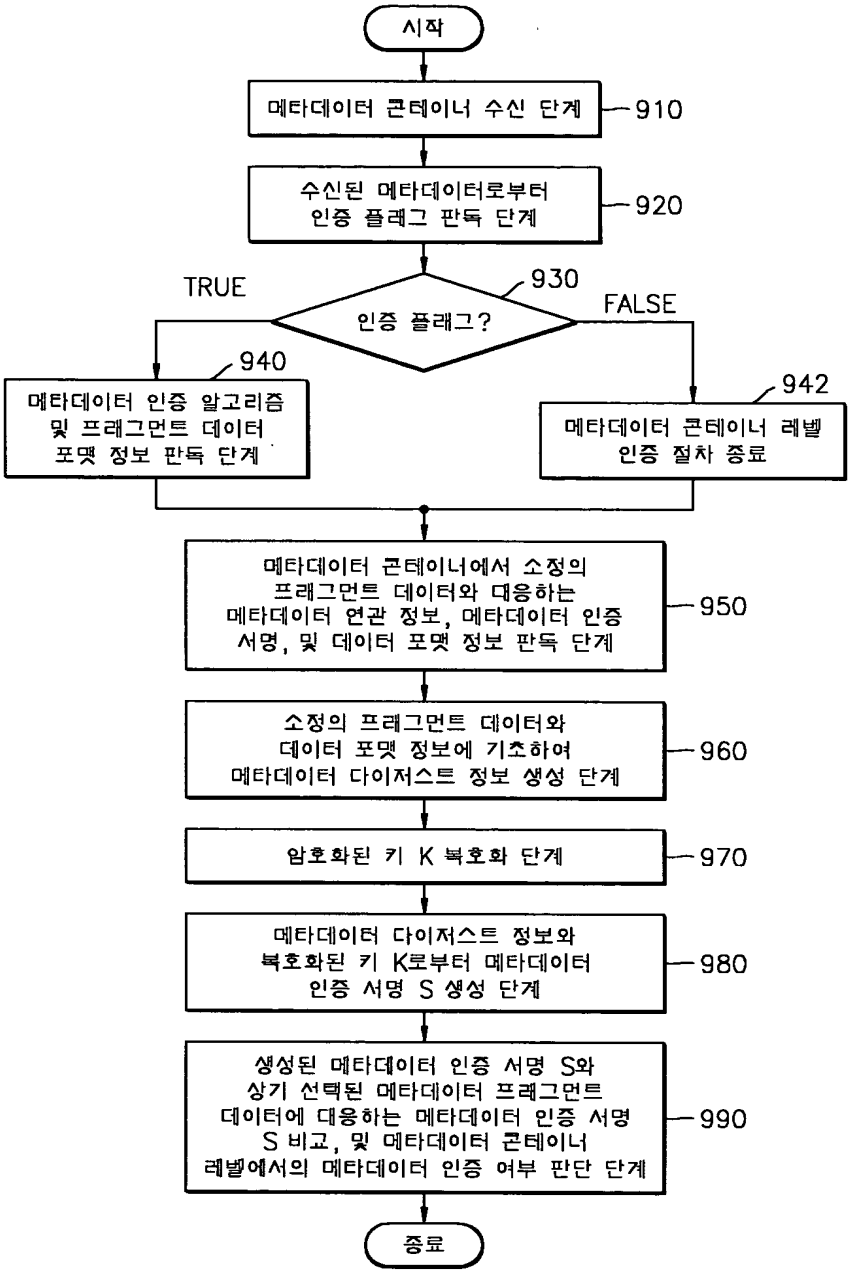
【도 7】


시작
메타데이터 콘테이너 수신 단계
710
수신된 메타데이터로부터 인증 플래그 판독 단계
720
730
인증 플래그?
TRUE
FALSE
740
메타데이터 인증 알고리즘 포맷 데이터 정보 판독 단계
742
메타데이터 콘테이너 레벨 인증 절차 종료
메타데이터 콘테이너에서 소정의 프레그먼트 데이터와 대응하는 메타데이터 연관 정보 판독 단계
750
소정의 프레그먼트 데이터와 데이터 포맷 정보에 기초한 메타데이터 다이저스트 정보 생성 단계
760
생성된 메타데이터 다이저스트 정보와 소정의 프레그먼트 데이터에 대응하는 상기 판독된 메타데이터 다이저스트 정보 비교, 및 메타데이터 콘테이너 레벨에서의 메타데이터 인증 여부 판단 단계
770
종료

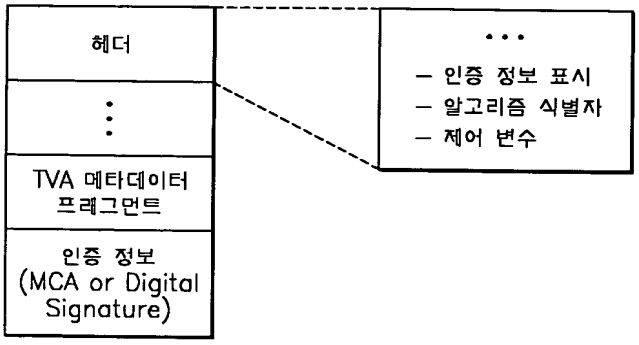
【도 8】


시작
전송되는 메타데이터를 소정의 세그먼트 단위로 분할하여 복수개의 프래그먼트 데이터를 생성하는 단계
810
생성된 프래그먼트 데이터들 중 소정의 프래그먼트 데이터 선택 단계
820
선택된 프래그먼트 데이터로부터의 메타데이터 다이저스트 정보 생성 단계
830
생성된 메타데이터 다이저스트 정보와 암호화 키를 사용한 메타데이터 인증 서명 생성 단계
840
선택된 프래그먼트 데이터와 대응하는 메타데이터 다이저스트 정보, 메타데이터 인증 서명, 및 선택된 프래그먼트 데이터의 데이터 포맷 정보를 포함하는 메타데이터 콘테이너 생성 및 전송 단계
850
종료

【도 9】


시작
메타데이터 콘테이너 수신 단계
910
수신된 메타데이터로부터 인증 플래그 판독 단계
920
930
인증 플래그?
TRUE
FALSE
940
메타데이터 인증 알고리즘 및 프래그먼트 데이터 포맷 정보 판독 단계
942
메타데이터 콘테이너 레벨 인증 절차 종료
메타데이터 콘테이너에서 소정의 프래그먼트 데이터와 대응하는 메타데이터 연관 정보, 메타데이터 인증 서명, 및 데이터 포맷 정보 판독 단계
950
소정의 프래그먼트 데이터와 데이터 포맷 정보에 기초하여 메타데이터 다이제스트 정보 생성 단계
960
암호화된 키 K 복호화 단계
970
메타데이터 다이제스트 정보와 복호화된 키 K로부터 메타데이터 인증 서명 S 생성 단계
980
생성된 메타데이터 인증 서명 S와 상기 선택된 메타데이터 프래그먼트 데이터에 대응하는 메타데이터 인증 서명 S 비교, 및 메타데이터 콘테이너 레벨에서의 메타데이터 인증 여부 판단 단계
990
종료

【도 10】


헤더
TVA 메타데이터 프레그먼트
인증 정보 (MCA or Digital Signature)
• • •
– 인증 정보 표시
– 알고리즘 식별자
– 제어 변수

【도 11】

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP:Header>
    <wssec:credentials xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SigningCertificate">
      <ds:X509Data>
        <ds:X509Certificate>MIIH1zCCBr+gAwIBA...</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </wssec:credentials>
  <wssec:integrity xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI=""/>
          <ds:Transforms>
            <ds:Transform Algorithm="http://schemas.xmlsoap.org/2001/10/security
              #RoutingSignatureTransform"/>
            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>aiYECAxNqK2PivQaRweWajXup5zJa...</ds:SignatureValue>
      <ds:KeyInfo>
         <wssec:licenseLocation>
           #SigningCertificate
        </wssec:licenseLocation>
      </ds:KeyInfo>
       </ds:Signature>
  </wssec:integrity>
 </SOAP:Header>
<SOAP:Body>
  <TVAmetadataFragment>
    <ProgramInformation>.....</ProgramInformation>
        <enc:EncryptedData>
      <enc:EncryptionMethod Algorithm="xxx_algorithm"/>
      <ds:KeyInfo>.....</ds:KeyInfo>
      <enc:CipherData>
        <enc:CipherValue>9osy8Tw2+HcSHftHg...</enc:CipherValue>
      </enc:CipherData>
    </enc:EncryptedData>
    <enc:EncryptedKey>
          <enc:EncryptionMethod Algorithm="yyy_algorithm"/>
      <ds:KeyInfo>
       <ds:KeyName>Public/Private Key for TVA metadata</ds:KeyName>
        </ds:KeyInfo>
        <enc:CipherData>
          <enc:CipherValue>CCBPowCwYDVR0PBA...</enc:CipherValue>
            </enc:CipherData>
      </enc:EncryptedKey>
  </TVAmetadataFragment>
 </SOAP:Body>
</SOAP:Envelope>
```